Literature Review

Timed Automata has been studied since its introduction in the late 1980s. It was originally introduced by Alur and Dill as a formalism to model asynchronous real-time systems. A timed automaton is a finite automaton - a finite set of locations and transitions coupled with a finite set of real-valued variables called clocks. These clock variables evolve continuously at a unit rate and appear in guards of transitions in timed automata. In this section, we review some of the important results in this area. Here the syntax of timed automata allows clock values to be reset after executing a transition. A state of timed automaton is a pair of location and clock valuation where the state is called a 'corner state' if its clock valuation assigns integer values to all its clock variables. The formal definition of timed automata allows one to specify a special set of states called the set of final states.

A timed move is a pair (t, a) which represents the action: wait for the time duration-**t** before executing the transition with label **a**. A run of a timed automaton is an alternating sequence of states and timed moves <s0, (t1, a1), s1, . . . , sn> such that for every positive integer $i \leq n$, the timed move (ti , ai) is enabled in the state $si-1$ and the state si is reached after executing the timed move (ti , ai). An infinite run r = <s0, (t1, a1), . . .> is defined analogously. We say that a run <s0, (t1, a1), s1, . . . ,sn> is accepting if the state $sn \in F$. The language recognised by a timed automaton is the set of accepting runs. In their seminal paper, Alur and Dill showed that languages recognised by timed automata are closed under union and intersection, but not under complementation.

The reachability problem is to determine whether or not some target location is reachable. The reachability problem of a timed automaton is: given an initial state $s \in S$, decide whether a final state is reachable from the initial state, i.e., whether there exists an accepting run starting from s. Alur and Dill showed that the reachability problem for timed automata is decidable and that it is PSPACE-complete. They established PSPACE-membership of the reachability problem using a finitary abstraction—the so-called region graph—of timed automata, whose size is exponential in the size of the timed automaton. For the PSPACE hardness result they showed that for a linear-space Turing machine M and an input word w of length n, there exists a timed automaton T with 2n + 1 clocks, such that the language of T is nonempty iff the machine M accepts w.

Courcoubetis and Yannakakis later tightened the PSPACE-hardness result by showing a reduction from the acceptance problem for linear-space Turing machine to the reachability problem of timed automata with only three clocks. Minimum and maximum reachability-time problems were shown to be decidable by Courcoubetis and Yannakakis. It was shown to be PSPACE-complete by Alur, Courcoubetis, and Henzinger and Kesten et al. An efficient algorithm to solve the minimum reachability-time problem on timed automata appeared in 'Minimum-time reachability for timed automata', where the initial state was restricted to corner states.

In 2008, the concept of Alternating Timed Automata was brought up by Sławomir Lasota and Igor Walukiewicz. An alternating timed automaton is a mix of both timed automata and alternating finite automata. That is, it is a sort of automata which can measure time, and in which there exists universal and existential transition. ATAs are more expressive than a timed automaton. In fact, one clock alternating timed automaton (OCATA) is the restriction of ATA allowing the use of a single clock. OCATAs allow to express timed languages which can not be expressed using timed-automaton.

In conclusion, timed automata has been an interest of study since the early 1980s and helped in the design of time-constrained systems. Also, one of the future works is to develop fully automated optimization techniques that could maximally reduce the number of states and clocks while flattening the Timed Automata patterns. These clocks and states have a significant impact on the performance of tools like UPPAAL. By fully comparing the expressiveness of timed process algebras and Timed Automata, we'll have a significant gain to support verifying real-time systems.

References

[1]. Rajeev Alur e David L. Dill. “A Theory of Timed Automata”. Em: *Theoretical Computer Science* 126 (1994), pp. 183–235.

[2]. *Timed automaton*: https://en.wikipedia.org/wiki/Timed_automaton.

[3]. Lasota, Sławomir; Walukiewicz, Igor (2008). "Alternating Timed Automata". *ACM Transactions on Computational Logic*. **9** (2): 1–26.

[4]. R. Alur and D. Dill. Automata for modeling real-time systems. In International Colloquium on Automata, Languages and Programming (ICALP), volume 443 of LNCS, pages 322 – 335. Springer, 1990

[5]. Patricia Bouyer. *Timed Automata and Extensions: Decidability Limits*. Invited talk, 5èmes Journées Systèmes Infinis (JSI’05), Cachan, France. Mar. de 2005.

[6]. Patricia Bouyer. *Timed Automata — From Theory to Implementation*. Invited tutorial, 6th Winter School on Modelling and Verifying Parallel Processes (MOVEP’04), Brussels, Belgium. 27 pages. Dez. de 2004.